Two-Step X-Ray Transit Identification: Bayesian Block Simplification and Sequential Machine Learning Techniques

Lauren Shen

Under the direction of

Dr. Vinay Kashyap and Dr. Rosanne Di Stefano

Harvard-Smithsonian Center for Astrophysics

Harvard University

Abstract

The transit method is versatile; it has been critical in not only the discovery of over four thousand Milky Way exoplanets but also the impressive discovery in galaxy Messier 51 of M51-ULS-1b, the first possible extragalactic planet. Thus, the transit method is key in continuing the search for such phenomena. However, current methods of detecting transits involve visual identification, take significant time, and can be prone to human error. Combined with the large amount of data available, these observations naturally point to the use of computational techniques to aid the transit method. In this work, a two-step development of a machine-learning model was proposed to automate transit identification. In the first step, a simplified light curve was generated using the Bayesian blocks algorithm. Then, time-series datasets containing sections of event lists (sorted depending on the presence of a transit) were created. A training dataset was created from a source in 47 Tucanae containing many example transits; a validation dataset was created from the transit of M51-ULS-1b as a prime example of an extragalactic planet. In the second step, a random forest model was trained, optimized, and evaluated: it performed with high accuracy and was able to find the exact point in time of the transit for M51-ULS-1b. The main merits of the model are (i) the simplicity of the data, as the only needed feature is time, and (ii) its ability to successfully learn from astrophysically different datasets. This method is unique because of its efficiency and applicability: it significantly focuses the approach to transit identification by reducing the time (from days to minutes) and possible errors involved in finding statistically significant transits and also allows astrophysicists to perform meaningful work without the need for an "intuition."

1 Introduction

The goal of discovering exoplanets (planets outside of the Solar System) has sparked interest in the scientific community ever since the first exoplanet was confirmed in 1992. [1] With over 5,000 exoplanets discovered in just the Milky Way in the past thirty years [2], such interest has recently been heightened with the detection of M51-ULS-1b, the first *extragalactic planet* candidate (from the Messier 51 galaxy). [3] This candidate was found through the transit method of detection, which involves the identification of characteristic decreases in the count rate (rate of photon arrival, with units of photons per second) that may be indicative of an exoplanet passing between the photon source and the point of observation. The fact that M51-ULS-1b was discovered through the transit method from analysis of X-ray data from the Chandra X-ray observatory indicates three key points. First, such a detection shows that it is probable for other planets and potentially interesting cosmic phenomena to also be detected outside of the Milky Way. Second, the use of X-ray data in such a detection indicates that Chandra may contain a rich source of data in regard to the search for exoplanets. Third, when taken together with the fact that the majority of the confirmed exoplanets in the Milky Way were also discovered with the transit method [2], this method is very promising in the context of future exoplanet discoveries. Thus, this research focuses on identifying transits in X-ray data with the eventual goal of discovering exoplanets in galaxies other than the Milky Way.

While the applicability of the transit method of detection is evident, several issues arise when considering the search for exoplanets on a larger scale. Even solely considering X-ray data from Chandra, the Chandra Source Catalog contains over 36 terabytes of data [4] from over three hundred thousand unique sources. The manual process for identifying transits in light curves (plots of the count rate over time) becomes difficult if one is to perform a thorough search of the catalog. The immense amount of data available makes this manual process both impracticable in terms of time consumption and risky in terms of potential mistakes.

Although it is difficult to manually approach the search for exoplanets, the fact that a significant amount of X-ray data is available and that characteristics of desired transits are known point to implementing a machine-learning model as a promising approach. In fulfilling the goal of creating a machine-learning model that is able to identify transits in various light curves, a pre-screening method involving Bayesian blocks [5] was first implemented with the aim of identifying the optimal segmentation of the times of arrival for the photons. In other words, the Bayesian blocks algorithm is able to find the most significant points in time for each light curve, which allows for significant simplification for a light curve while still preserving its most important characteristics. Then, with this new set of light curves, those that still contain transits after simplification are noted and added to the training dataset as examples of the type of transit the machine learning model should detect. Thus,

after running Bayesian blocks on a significant number of light curves and manually identifying a representative dataset of transits, a training dataset is generated, from which the machine learning model can learn. In particular, random forest classifiers, which are helpful in our classification of which light curves contain dips, were implemented.

Machine-learning techniques have been applied to the problem of identifying what particular source a light curve generated from Chandra data shows [6] or to the similar question of detecting exoplanets through analysis of light curves. For example, a previous work [7] utilized the time-series analysis library TSFresh to extract the most significant features from light curves generated from Kepler data and then implemented a machine-learning classifier model to detect exoplanets through the transit method. However, the use of Bayesian blocks as a pre-screening tool and the specific machine-learning algorithms applied in this research have not been previously implemented for the purpose of identifying particular transits in various light curves from the Chandra X-ray Observatory data. In particular, previous methods rely on features of light curve data that are not present in X-ray data, which must be used if cosmic phenomena in other galaxies are to be detected. Such features may involve the periodicity of exoplanets (unlikely to be observed at larger distances) as they orbit their stars or the relatively high number of photons received (there are significantly fewer numbers of photons received from more distant sources). Thus, this research seeks to address the following questions:

1. How can Bayesian blocks be used as a pre-screening tool to aid in transit identification?
2. How can transits be consistently identified in light curves generated from Chandra X-ray data using machine learning models such as random forest classifiers?
3. How can a machine learning model trained on a particular dataset be applied to data generated from astrophysically diverse sources? If such a generalization can be achieved, the applicability of this model would be greatly enhanced.

2 Background

2.1 Chandra X-ray Data and Light Curves

NASA's Chandra X-ray Observatory was launched in 1999 and was specially designed to collect X-ray data. Thus, its data is often used in analyzing the types of sources that give off such high-energy radiation, such as galaxy clusters, black holes, or X-ray binary star systems. Because X-ray data is high in resolution and low in background noise, light curves generated from such data are well-suited for transit searches. An example of a transiting exoplanet is shown in **Figure 1a**, with a note to how the dip begins the moment the exoplanet passes in between the point of observation and the star.

2.2 Characteristics of Transits

Several important characteristics of exoplanet transits are demonstrated in **Figure 1b**, which shows the transit of the first extragalactic planet candidate, M51-ULS-1b. The transit is symmetric, decreases significantly to no counts, and also returns to its average value before the dip. Additionally, it is relatively short (around 15 ks). Thus, this dip is a prime example of the desired type of transit to find, which is why data from this light curve has been used in validating the machine learning model.

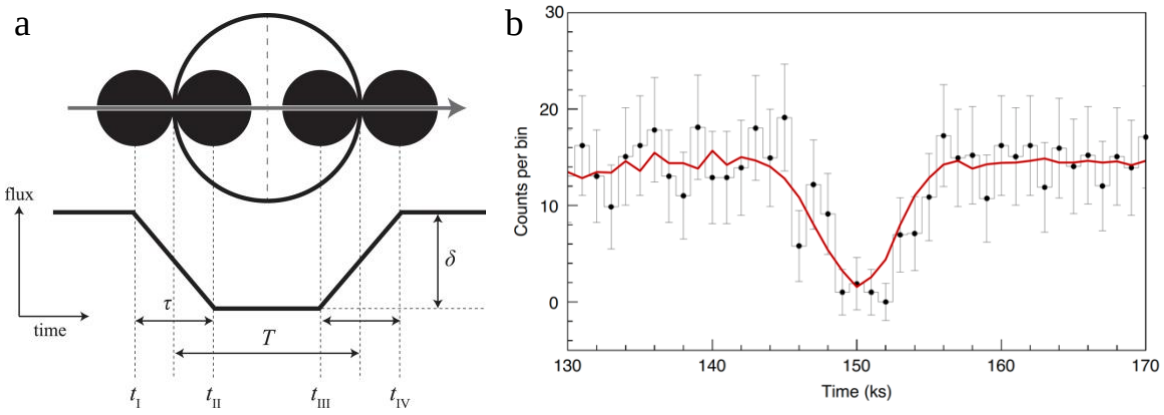


Figure 1. a) Example of exoplanet transit [8]. b) Extragalactic planet transit example for M51-ULS-1b. Note the presence of the mentioned notable characteristics of transits [3].

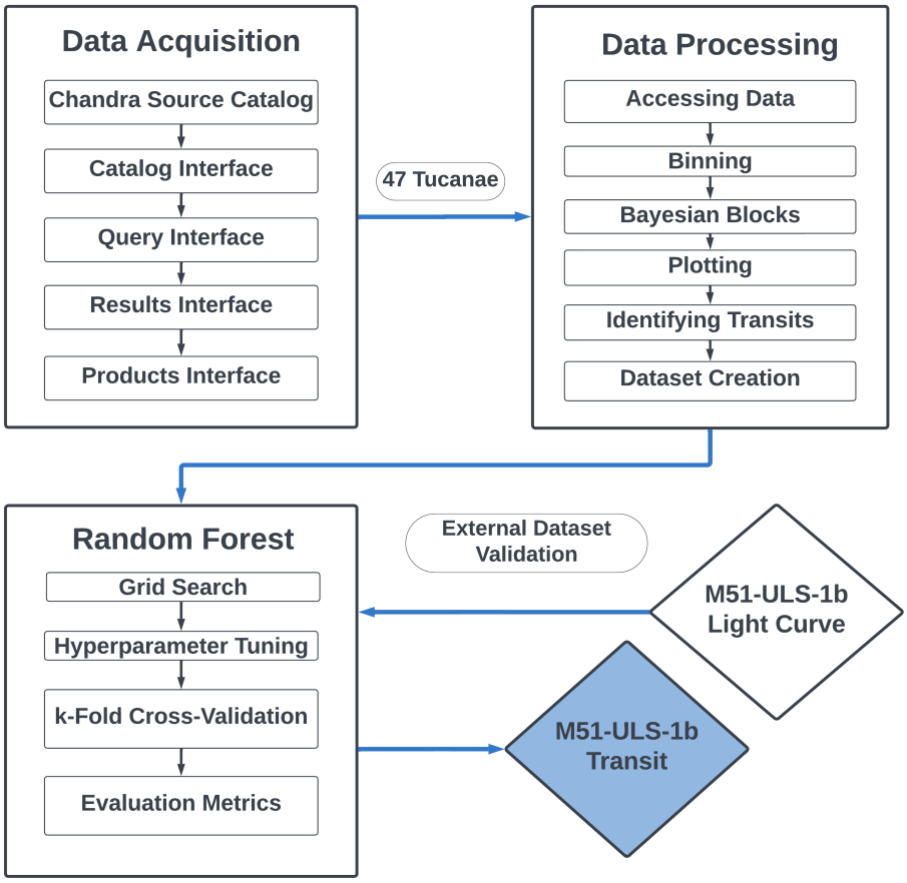
3 Methods

After data was acquired from the Chandra Source Catalog, it was preprocessed and the Bayesian blocks-simplified light curve was plotted over the light curve to aid in identifying when transits occurred. Thus, a time-series dataset with indication of whether or not each section of time contained a transit was able to be created. On this data, the random forest classifier model was then trained, optimized, and finally evaluated on both the test dataset and a separate dataset. **Figure 2** shows the process of the methods used in this project.

3.1 Data Acquisition

All data was acquired from the Chandra Source Catalog using the CSCview application. To navigate the application, four interfaces were used: the catalog, query, results, and products. In the catalog, the current database of Chandra was selected, which moved the application to the query interface, where a region was searched for either by name or by coordinates. The names used to find the sources in 47 Tucanae and Messier 51 can be found in **Table 1.** Then, in the results interface, the event list and source region files were selected for a particular source, as these were the only files needed to produce the desired type of light curves. The event list contains the arrival time of each photon, and the source region contains data regarding the background count rate of each source, which aids in later considerations of the significance of each light curve. Finally, in the products interface,

the corresponding event list and source region files were downloaded for each needed observation of the



source.

Figure 2. The flowchart showing the development and evaluation of the machine learning model.

Table 1. Data from two sources in 47 Tucanae and Messier 51, respectively. The dataset from 47 Tucanae was used to train the model and thus is significantly larger than the dataset from M51, which will be used in the external validation step.

Location	Source Name	Number of Windows		
		Transits Absent	Transits Present	Total
47 Tucanae	2CXO J002400.9-720453	19448	3259	22707
Messier 51	2CXO J132943.3+471134	1764	240	2004

3.2 Data Preprocessing

The Bayesian blocks algorithm was run and plotted over various light curves to aid in the process of visually identifying dips. Then, after dips of shorter than 20 ks had been identified (as dips longer than 20 ks are less likely to be exoplanet transits), data regarding the light curve and dip were saved to be used in the machine learning dataset. A program was written to automate the process of producing light curves and running Bayesian blocks after downloading the data in a .tar file from the

Chandra Source Catalog. Necessary Python libraries were imported to extract data from the .tar file, and a shell script was used along with several functions to process and plot the data. After processing, the needed information about each graph was temporarily stored as a pandas dataframe and then permanently in a .csv file, which allowed for the preprocessed data for each source to be saved for future steps.

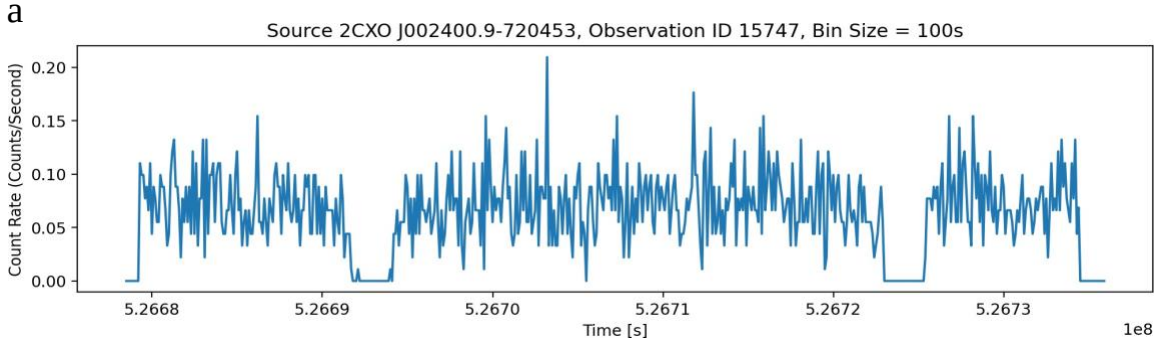
3.2.1 Accessing Data

The following Python libraries were imported to unzip, access, and manipulate the data: astropy, csv, matplotlib, numpy, os, pandas, shutil, subprocess, and tarfile. Files were unzipped from .tar to .gz to .fits file formats. Each observation ID (obsID) and file name was stored in a pandas dataframe, with care taken to ensure that the .fits files were stored in the correct event list and source region pairs. A shell script then made use of the dmextract and dmlist tools to access the arrival time and energy level in electron volts (eV) for each photon. Specifically, the photons of energy levels between 300 to 7000 eV were extracted for use in making light curves.

3.2.2 Binning

After the necessary data for each observation was extracted, the next steps aimed to generate the light curve. The creation of light curves relies on a process called *binning* for the photons, requiring only the arrival time for each photon. A bin size is first set to a particular length of time (usually 500 s). The total number of photons received during each time interval of the bin length is then found. The average count rate, or photons received per second, over this time interval is calculated, and after binning all of the photons for a particular observation, a plot of the count rate vs. the overall observation time is created to represent the light curve. Selecting an appropriate bin size is important. The following figures show how changing the bin size affects the complexity of the light curve. The example light curve shown is from one of the sources in 47 Tucanae on which the model was trained. Note how as the bin size gets larger for **Figures 3a-3c**, the light curve becomes less complex.

a



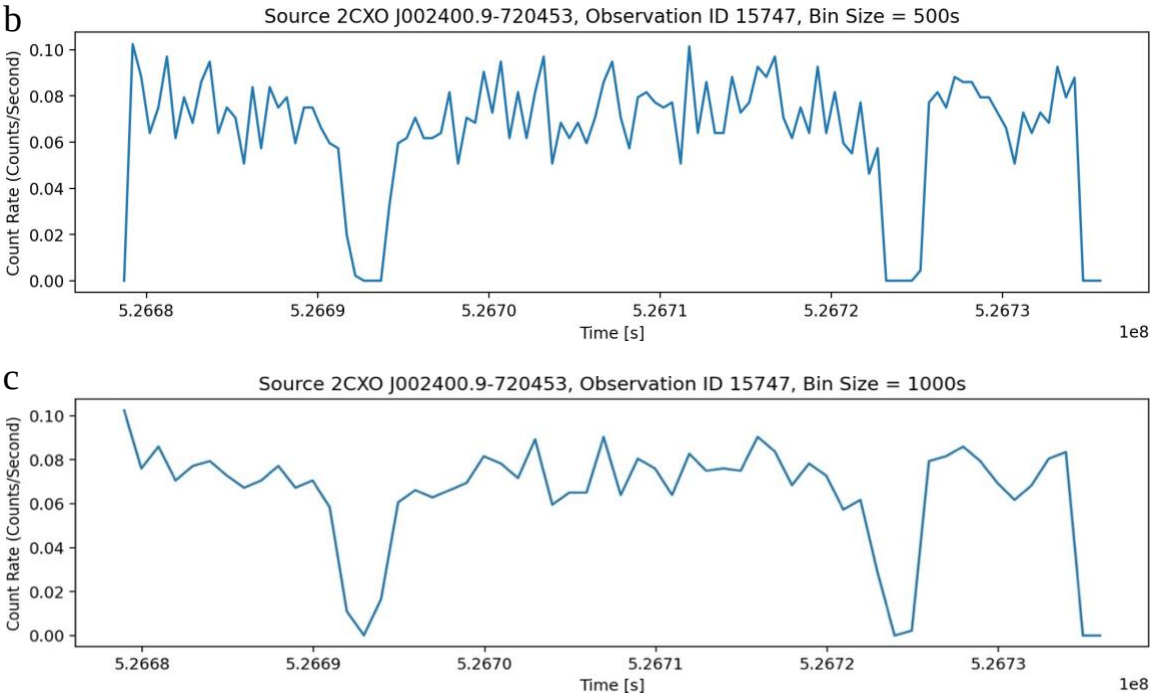


Figure 3. Light curve for source in 47 Tucanae, shown for bin size of a) 100 s, b) 500 s, and c) 1000 s.

3.2.3 Bayesian Blocks Algorithm

A function was constructed to run the Bayesian blocks algorithm, which needed to access only the data of the arrival times of the photons. After setting a significance value, p_0, the algorithm identified a number of *edges*, or points in time that were deemed to be significant in the context of the light curve. Essentially, the algorithm finds the most appropriate bin size for each segment of the light curve. The average count rate was then calculated for each duration of time defined by the edges, and the edges and count rates were saved as lists to the pandas dataframe that temporarily stores the collected data for each observation.

3.2.4 Plotting

A function was written to plot light curves with the purpose of aiding dataset creation. Light curves were generated with the Bayesian blocks-simplified light curve graphed over the broadband light curve, with each edge of the simplified curve labeled with its index so that accessing the time for each edge would be easier. Each horizontal axis shows the time (s) while each vertical axis shows the count rate (photons/second).

3.3 Dataset Creation

3.3.1 Accessing Transit Edges

The purpose of plotting and labeling each step of the Bayesian blocks algorithm becomes evident in this step, where a function was written to take in the indices of the edges of the dip(s) found in a light curve and output the actual points in time of the edges. The duration of each dip was also recorded if the graph of the Bayesian blocks-generated light curve had not shown the duration to be obviously less than 20 ks. The transit edges and duration(s) were then saved to the temporary pandas dataframe. If no dips were present, this observation was noted for the light curve as well.

3.3.2 Creating Input and Output Data

A particular format of data was needed for the machine learning model to function properly. The input data consisted of uniformly fixed *sliding windows* that captured sequential sections of time in the light curve, while the output data consisted of either 0 or 1, indicating the absence or presence of a dip in that section of the light curve, respectively. For this dataset, the window size was set to 250 points in time because such a size ensured that there would be a sufficient amount of train and test data. The windows allow a significant amount of data to be generated from the event list for a single light curve as follows: the first window contains the arrival times of the first photon through the 250^{th} photon of the light curve; the second window contains the arrival times of the second photon through the 251^{th} photon; and so on. Thus, as the window slides through the event list, each row of the input data contains a 250-element list of consecutive arrival times of photons, and the objective is that the machine learning model will be able to learn patterns in each list, associate such patterns with the absence or presence of transits, and eventually be able to predict whether or not a given list of arrival times contains a transit.

The process described above produces the input and output data for only a single light curve. To produce a larger dataset from which the model can learn, the data is extracted for multiple light curves and then concatenated. In this work, the model was trained on data from a source in the globular cluster 47 Tucanae, chosen because it contained confirmed examples of transits (specifically, of rotating binary star systems) and thus would be a reliable set of data on which the model could train. Data was also collected from the source in galaxy Messier 51 that had allowed for the discovery of the exoplanet candidate M51- ULS-1b, as this source has been thoroughly analyzed and is known to contain a transit. The hope is to train the model on the data from 47 Tucanae and then verify the model using the data from M51. Information regarding this data is shown in **Table 1.**

3.4 Machine Learning

3.4.1 Random Forest Implementation

Random Forests are an ensemble learning technique commonly used for classification. They involve an ensemble of decision trees, where each tree learns from a random subset of the data to predict the classification. Then, the predictions of all of the decision trees in the model are combined

to produce a final classification. The RandomForestClassifier and train_test_split modules were imported from the sci-kit learn package for the implementation of the model. The input and output data were first converted to numpy arrays, and then the input data was flattened in preparation for running the model. Then, the dataset was split into train and test subsets, with the test subset making up 20% of the data.

3.4.2 Grid Search and Hyperparameter Tuning

To improve the random forest model, hyperparameter tuning was performed through a grid search. Hyperparameters are parameters that are set before the model is allowed to learn from the dataset and determine exactly how the model is trained. For this random forest model, the hyperparameters included the maximum number of trees in the forest, the maximum depth of the trees, and the minimum number of samples required to split or be at a leaf node. Grid search is a method by which the best combination of hyperparameters is found for a particular machine-learning model. A grid of possible hyperparameter values is defined and the model is trained and evaluated for each combination, allowing the optimal set of hyperparameters to be selected for the model. In particular, grid search was chosen over other methods of hyperparameter optimization (such as random search or Bayesian optimization) for its exhaustiveness (as it tries every possible combination) and reproducibility (as it produces the same results each time for the same given circumstances).

3.4.3 k-Fold Cross-Validation

After the most optimal hyperparameters were determined for the random forest model, performance was further improved and evaluated through k-fold cross-validation. This method involves splitting the dataset into k (k = 6 in this work) subsets, or folds, of roughly equal size and then training and evaluating the model k times, each time using a different fold as the test set and the rest of the data as the training set. The evaluation metrics for each fold are calculated and then compared to one another to better understand the performance of the model as a whole.

3.4.4 Evaluation Metrics

The following evaluation metrics were used to evaluate the model's performance:

- Accuracy is the ratio of the number of correctly predicted samples to the total number of samples in the dataset. It was chosen as an appropriate evaluation metric because of its interpretability in this context.
- Precision is the ratio of true positives (transits identified by the model to be transits) to the sum of true positives and false positives (non-transits identified to be transits). It is a useful metric because a goal of this work is to streamline the transit identification process, and thus false positives must be considered and minimized for the model to be effective.
- Recall is the ratio of true positives to the sum of true positives and false negatives (transits

identified to be non-transits). Similarly, the number of false negatives should be minimized; otherwise, the model could miss potentially important transits.

- The F1 score is the harmonic mean of precision and recall, providing a more general idea of the model's performance. The following formula shows how the F1 score is calculated.

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

where,

$$precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

- The ROC AUC (Receiver Operating Characteristic Area Under the Curve) score is the area under the curve of the graph of the true positive rate against the false positive rate and indicates how well the model is able to distinguish between positive and negative samples.
- The confusion matrix is a table that indicates the number of true positives, true negatives, false positives, and false negatives, all of which are essential to evaluating the performance of a binary classification model.

3.4.5 External Dataset Validation

After the model was trained, optimized, and evaluated on the data from 47 Tucanae, it was then fed input data from the light curve that had allowed for the discovery of the exoplanet candidate M51-ULS-1b. In this way, the model was able to not only be evaluated on an external dataset but also be tested for future applicability, as its eventual purpose is to be able to detect transits in any given input data of the correct formatting and window size.

4 Results and Discussion

4.1 Significance of p_0 and Bayesian Blocks Simplification

The following figures show not only how the Bayesian blocks algorithm is able to simplify a light curve, but also how the significance value p_0 affects the degree of simplification. The example light curve shown is from a source in the galaxy IC 10. Each figure shows how the simplified graph still preserves the important characteristics of each light curve, but note that **Figure 4a-4c** show how the light curve becomes more simplified as the p_0 value is set to lower values, indicating higher thresholds for significance.

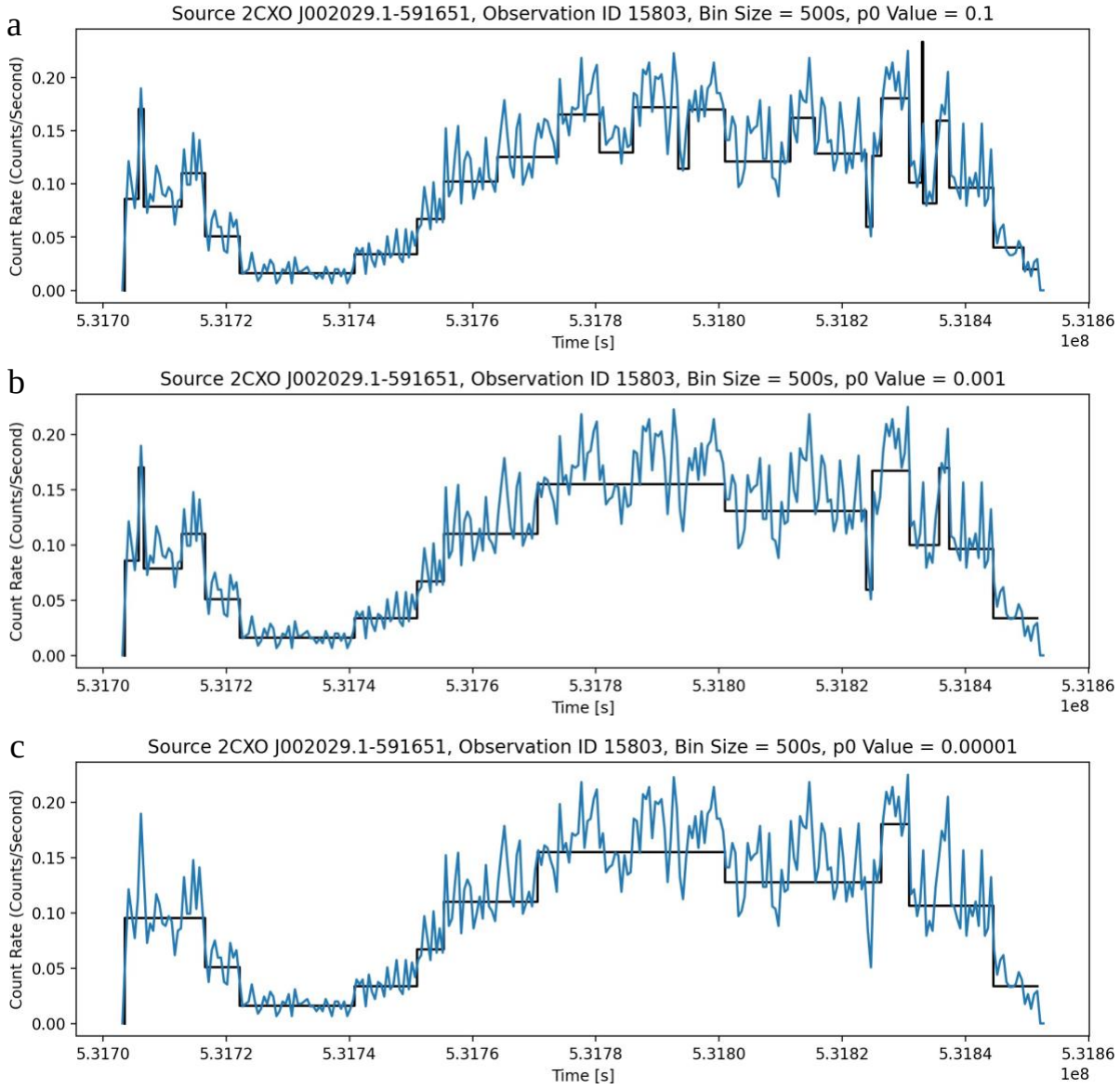


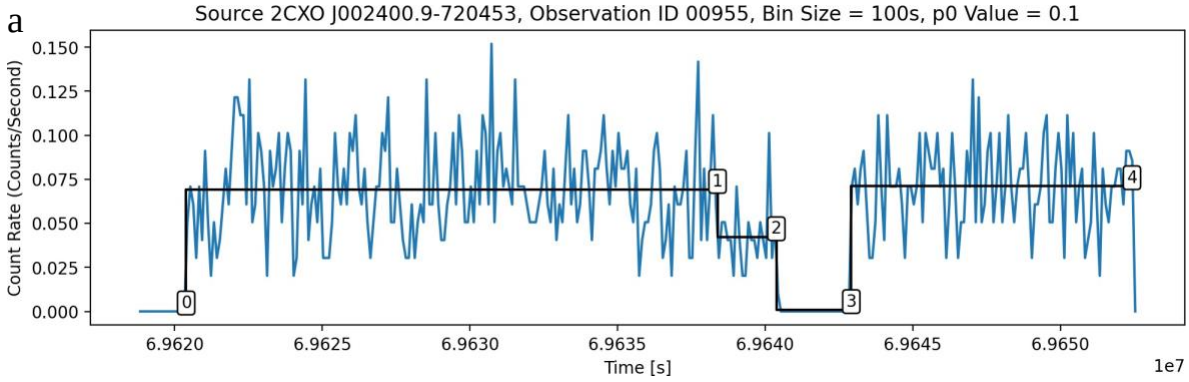
Figure 4. a) Light curve for source in IC 10 with 29 edges when p_0 is set to 0.1. b) Light curve for source in IC 10 with 18 edges when p_0 is set to 0.001. c) Light curve for source in IC 10 with 12 edges when p_0 is set to 0.00001.

The Bayesian blocks algorithm was successful in simplifying light curves from 47 Tucanae. Example graphs of the results are shown in **Figures 5a-5e**, where the labels were then used in accessing the transit edges to create the first dataset. A summary of the transit edges identified by the algorithm shown in **Table 2.**

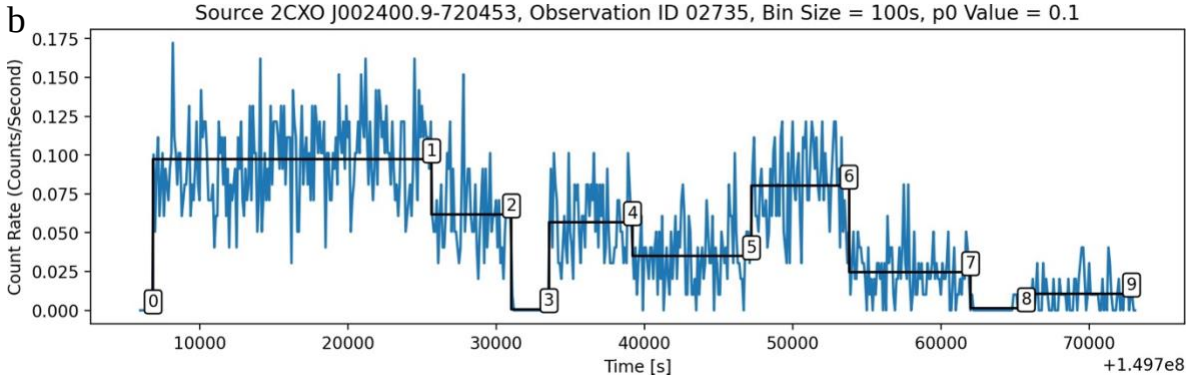
Table 2. Edges of transit(s) in each figure.

Figure	Transit Edges
Figure 5a	(2, 3)
Figure 5b	(2, 3), (7, 8)
Figure 5c	(1, 2), (3, 4)
Figure 5d	(2, 3), (5, 6)
Figure 5e	(2, 3)

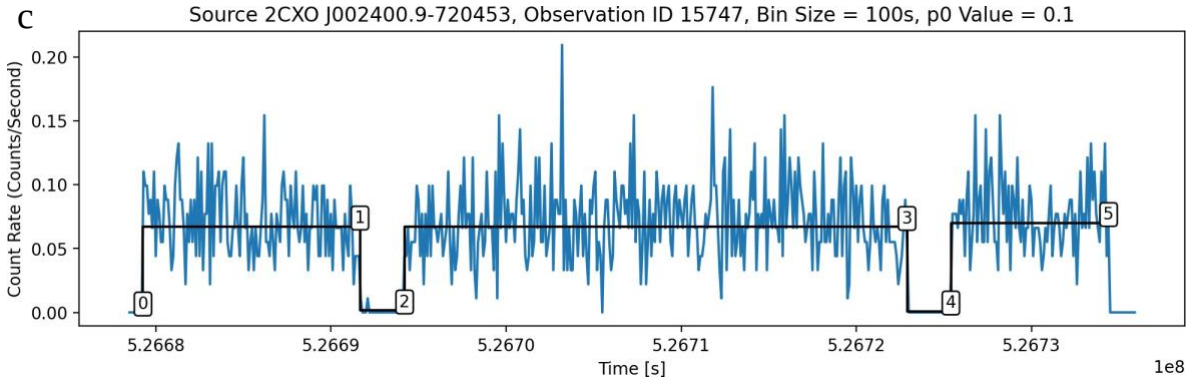
a


Source 2CXO J002400.9-720453, Observation ID 00955, Bin Size = 100s, p0 Value = 0.1
Count Rate (Counts/Second)
0.150
0.125
0.100
0.075
0.050
0.025
0.000
6.9620
6.9625
6.9630
6.9635
6.9640
6.9645
6.9650
Time [s]
1e7

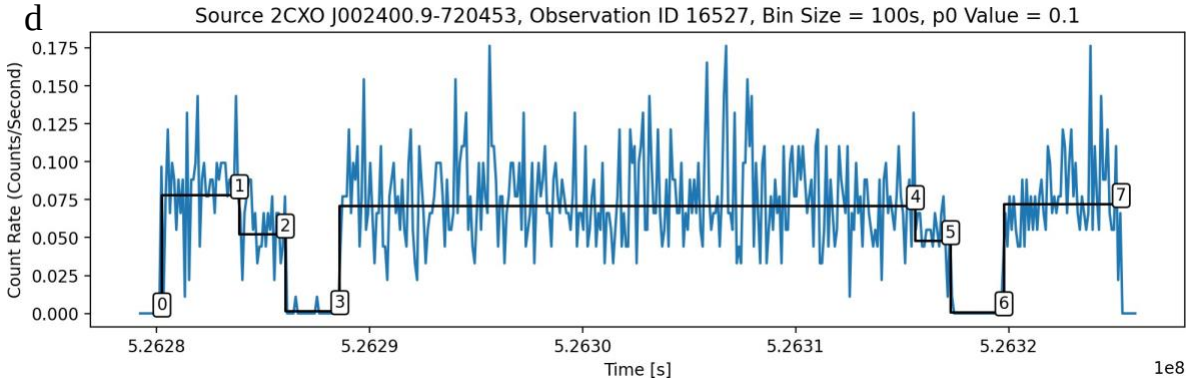
b


Source 2CXO J002400.9-720453, Observation ID 02735, Bin Size = 100s, p0 Value = 0.1
Count Rate (Counts/Second)
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
10000
20000
30000
40000
50000
60000
70000
Time [s]
+1.497e8

c


Source 2CXO J002400.9-720453, Observation ID 15747, Bin Size = 100s, p0 Value = 0.1
Count Rate (Counts/Second)
0.20
0.15
0.10
0.05
0.00
5.2668
5.2669
5.2670
5.2671
5.2672
5.2673
Time [s]
1e8

d


Source 2CXO J002400.9-720453, Observation ID 16527, Bin Size = 100s, p0 Value = 0.1
Count Rate (Counts/Second)
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
5.2628
5.2629
5.2630
5.2631
5.2632
Time [s]
1e8

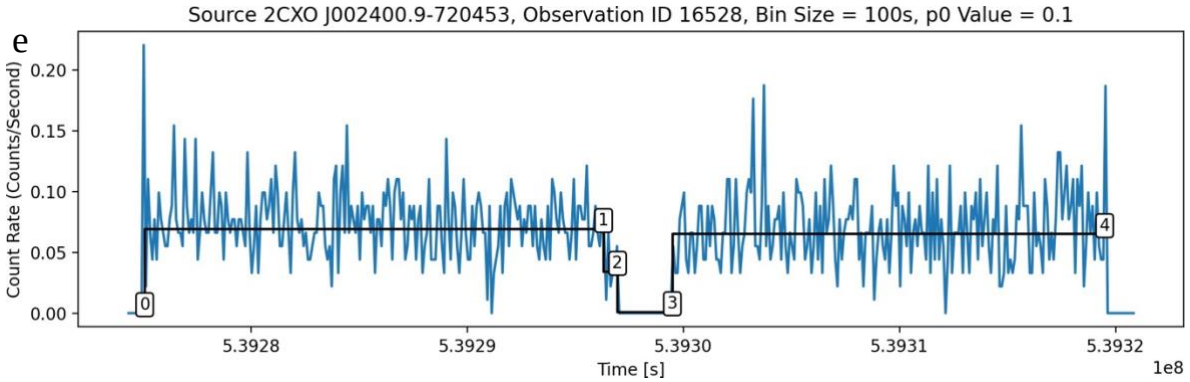


Figure 5. Bayesian blocks-simplified light curve for a) ObsID 00955, b) ObsID 02735, c) ObsID 15747, d) ObsID 16527, and e) ObsID 16528 of source from 47 Tucanae.

4.2 K-Fold Cross-Validation

4.2.1 Confusion Matrices

Six confusion matrices were produced from the six subsets of data that the model was trained and evaluated on as a result of the k-folds cross-validation, and a representative one is shown in **Figure 6**. The results from the other confusion matrices are shown in **Table 3**.

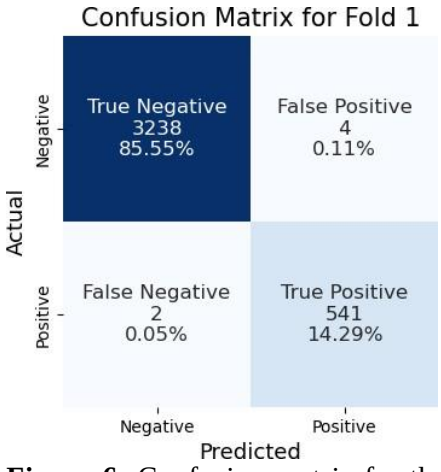


Figure 6. Confusion matrix for the first fold.

The model performs very well, as there is a high number of correct classifications for both positive and negative cases and a significantly low number of incorrect predictions for each fold. Furthermore, the proportion of true negatives to true positives is reflective of the proportion of windows that do not contain transits to the ones that do. The model is also consistent across folds as it produces very similar results, even on different subsets of data. Although these confusion matrices are similar to the point where the differences are visually negligible, the existence of such differences shows that the model is performing well and learning from the dataset, rather than simply memorizing the input data. Overall, these observations of the confusion matrices show that the model will likely generalize well to new data and that the hyperparameters found by the grid search method were a good fit.

Table 3. Results from the six confusion matrices produced from the six-fold cross-validation. Note from the mean and standard deviation (SD) how the results for each fold are similar, showing consistency in the model's performance, yet have slight differences, indicating the model's ability to adapt to new data.

Fold	True Positive	True Negative	False Positive	False Negative
1	541	3238	4	2

2	554	3226	0	5
3	506	3275	3	1
4	547	3233	2	2
5	550	3230	2	2
6	548	3234	1	1
Mean	541	3239.33	2	2.17
SD	17.66	17.92	1.41	1.47

4.2.2 Evaluation Metrics

The good performance of the model on the confusion matrices also increases reliability when interpreting the other evaluation metrics, shown in **Table 4.** The fact that the model performs well on each individual fold and produces similar (but not identical) results, as seen by the high mean and low standard deviation, demonstrates that the model is promising as it is quite robust and not overly sensitive to variations in the training data.

Table 4. Performance of the model, as evaluated by five different metrics. The mean and standard deviation (SD) across the six folds are also shown.

Fold	1	2	3	4	5	6	Mean	SD
Accuracy	0.99841	0.99867	0.99894	0.99894	0.99894	0.99947	0.99889	0.00035
Precision	0.99266	1.00000	0.99410	0.99635	0.99637	0.99817	0.99627	0.00265
Recall	0.99631	0.99105	0.99802	0.99635	0.99637	0.99817	0.99605	0.00259
F1 Score	0.99448	0.99550	0.99606	0.99635	0.99637	0.99817	0.99616	0.00121
ROC AUC	0.99754	0.99552	0.99855	0.99786	0.99787	0.99893	0.99771	0.00118

4.3 External Dataset Validation

In addition to achieving such promising results on the training dataset, the model also accurately predicted the *exact* point in time of the transit of M51-ULS-1b. The following figures show how the light curve for M51- ULS-1b was processed and the transit was identified. First, the light curve is plotted (**Figure 7a**). Then, the Bayesian blocks algorithm is run and the simplified light curve is plotted over the light curve (**Figure 7b**). Finally, each of the edges that the Bayesian blocks algorithm identifies to be important is labeled and shown on the graph (**Figure 7c**). With the labels, the edges of the transit are more easily identifiable and accessible through a function that was constructed to take in the labels as input and to output the transit edges and length.

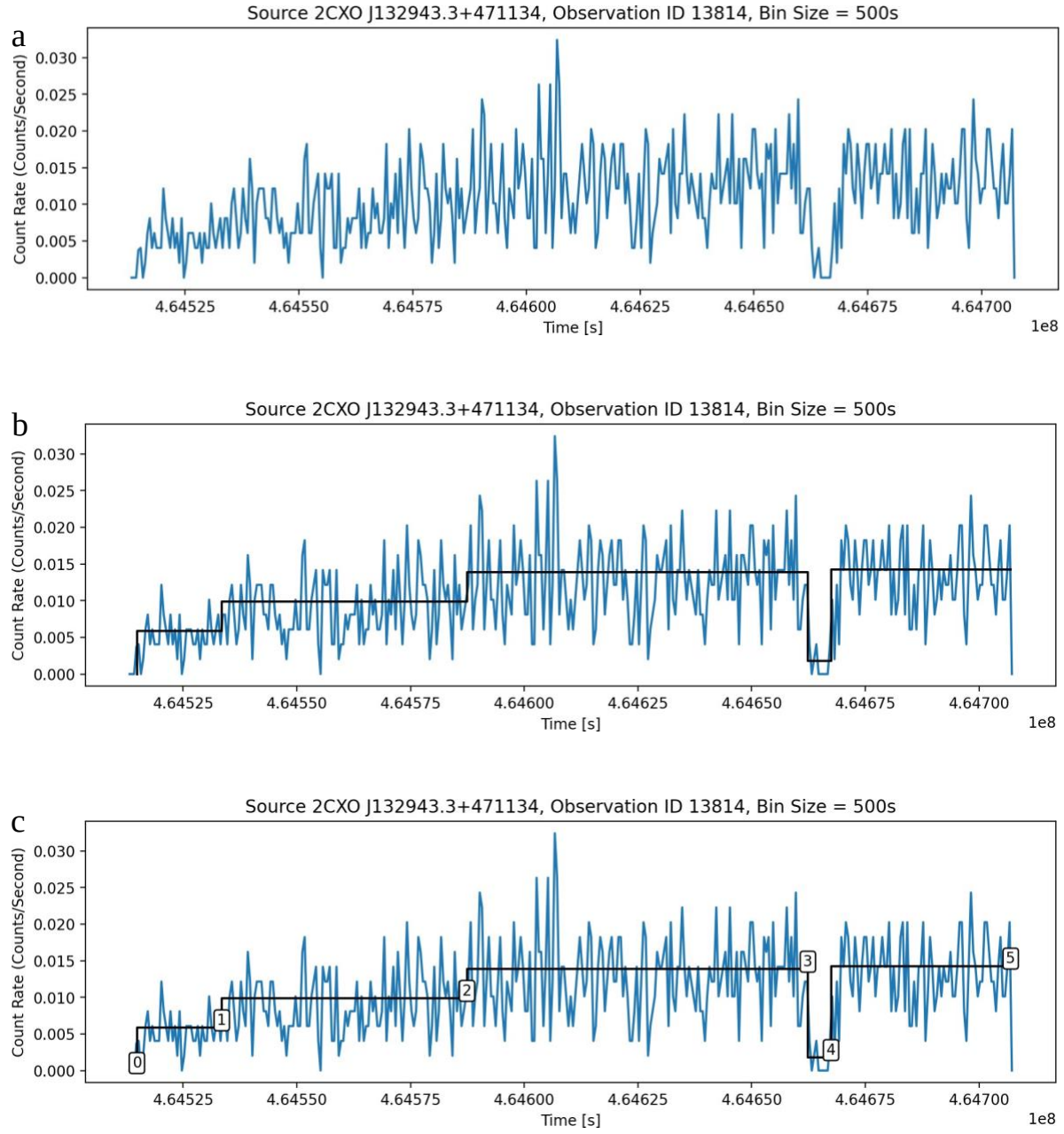


Figure 7. a) Light curve with bin size 500 s for M51-ULS 1b. b) Light curve overlain with simplified Bayesian blocks graph, with p_0 significance value set at 0.1. c) Light curve with simplified Bayesian blocks graph, shown with labels. This particular light curve with p_0 significance value has six edges.

This shows not only how this particular machine learning model is well-suited to the task of identifying transits in new data but also that machine learning can be a viable method of finding transits and thus cosmic phenomena. Additionally, even though the training and external validation datasets were obtained from very astrophysically-different sources, the model was still able to learn the necessary characteristics of transits in the time-series data from 47 Tucanae and apply them to

previously-unseen data in M51. This exciting realization shows the adaptability of the model and the flexibility that future astrophysicists will have in designing more efficient methods of transit detection.

5 Conclusions and Future Work

5.1 Conclusions

In this research, a two-step process was implemented to build a successful random forest classifier model that is able to accurately identify transits from time-series data. The first step involved the Bayesian blocks algorithm: when given the list of photon arrival times for a particular light curve, it identified the most significant points in time, effectively simplifying the light curve while still preserving the most distinctive characteristics. Then, after running this algorithm on transit data from 47 Tucanae and visually identifying points in time when transits occurred, a dataset consisting of time-series data and whether each segment of time contained a transit was generated. The second step involved training, optimizing, improving, and evaluating the random forest model. A grid search was conducted to find the optimal hyperparameters, and then the model's performance was evaluated using 6-fold cross-validation. The model performed extremely well across six metrics, with similar but not identical results, indicating its potential to be generalized to new data. Finally, the model was given the input time-series data from M51, which contained a single known transit, and was able to output its exact location, demonstrating great promise for its future applicability to new data. There are two main conclusions to draw from these exciting findings.

First, machine learning has not yet been applied to the specific issue of identifying transits in intergalactic X-ray data; previous work has focused more on implementing machine learning to infer the cause of the light curve from its shape, or has analyzed data only from within the galaxy. Thus, this work shows that it is possible to develop a machine learning model to accurately identify transits in this type of data.

Second, despite the significant astrophysical differences between the two sources of data used in this work, the model was still able to accurately identify transits in M51 after training on data from 47 Tucanae. The fact that the model is not limited by any astrophysical differences between the sources demonstrates its robustness and applicability.

5.2 Future Work

There are three main directions for future progress. First, the process of creating the dataset involved visual identification of transits. This manual process may not generalize well on a larger scale, so a viable direction for future work is to develop an algorithm that will work with the Bayesian blocks-simplified light curve to identify transits.

Second, the choice of a random forest classifier in this work was motivated by the fact that such

models have been used before in characterizing the sources shown in light curves. However, models other than random forest classifiers can and should be implemented to compare findings and performance. Direct next steps might include recurrent neural networks, as they are designed to work with sequential time-series data.

Third, the flexibility and simplicity of the model should also be stressed. This particular model will be most adept at identifying transits similar to the ones from 47 Tucanae due to the choice of the training dataset. However, it could easily be retrained on additional data to become more sensitive to different types of dips, demonstrating how astrophysicists may fine- tune the model to fit their particular needs. Additionally, other models that identify transits have relied on a wide variety of features in a light curve, with many requiring dimensionality reductions as a preprocessing step. While more features help to improve a model, the model in this work achieves high accuracy while needing only time-series data in addition to the binary classification of whether or not a transit is present, highlighting its simplicity.

To be clear, the purpose of this model is not to replace the human eye in finding exoplanets or other cosmic phenomena through the transit method. Rather, it aims to streamline the process of transit identification so that astrophysicists can spend more time focusing on sources that may yield more significant results. The hope is that the rapid advancement of machine learning models may be implemented to aid astrophysicists in the long-term search for exoplanets.

References

1. Wolszczan, A. and D.A. Frail, *A planetary system around the millisecond pulsar PSR1257 + 12.* Nature, 1992. **355**(6356): p. 145-147.
2. *NASA exoplanet archive*. 2023 [cited 2023 July 23]; Available from: https://exoplanetarchive.ipac.caltech.edu/docs/counts_detail.html.
3. Di Stefano, R., et al., *A possible planet candidate in an external galaxy detected through X-ray transit.* Nature Astronomy, 2021. **5**(12): p. 1297-1307.
4. *Chandra Source Catalog Release 2.0 (CSC 2.0)*. [cited 2023 July 9]; Available from: https://cxc.cfa.harvard.edu/csc/.
5. Scargle, J.D., et al., *STUDIES IN ASTRONOMICAL TIME SERIES ANALYSIS. VI. BAYESIAN BLOCK REPRESENTATIONS.* The Astrophysical Journal, 2013. **764**(2): p. 167.
6. Yang, H., et al., *Classifying Unidentified X-Ray Sources in the Chandra Source Catalog Using a Multiwavelength Machine-learning Approach.* The Astrophysical Journal, 2022. **941**(2): p. 104.
7. Thompson, S.E., et al., *A MACHINE LEARNING TECHNIQUE TO IDENTIFY TRANSIT SHAPED SIGNALS.* The Astrophysical Journal, 2015. **812**(1): p. 46.
8. Imara, N. and R. Di Stefano, *Searching for Exoplanets around X-Ray Binaries with Accreting White Dwarfs, Neutron Stars, and Black Holes.* The Astrophysical Journal, 2018. **859**(1): p. 40.